Journal of Process Control 21 (2011) 685–697

Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

Distributed model predictive control based on agent negotiation

J.M. Maestre, D. Muñoz de la Peña*, E.F. Camacho, T. Alamo

Dept. Ingeniería de Sistemas y Automática, Universidad de Sevilla, Sevilla, Spain

ARTICLE INFO

Article history:
Received 6 July 2010
Received in revised form
10 December 2010
Accepted 10 December 2010
Available online 5 March 2011

ABSTRACT

In this paper we consider the control of several subsystems coupled through the inputs by a set of independent agents that are able to communicate. We assume that each agent has access only to the model and the state of one of the subsystems. This implies that in order to take a cooperative decision, the agents must negotiate. At each sampling time agents make proposals to improve an initial feasible solution on behalf of their local cost function, state and model. These proposals are accepted if the global cost improves the cost corresponding to the current solution. In addition, we provide conditions that guarantee that the closed-loop system is asymptotically stable along with an optimization based design procedure that is based on the full model of the system. Finally, the proposed scheme is put to test through simulation.

1. Introduction

Traditionally, control theory has coped with information and timing constraints in a centralized fashion, that is, under the assumption that all the information is available at a single point at the right time. Unfortunately, there are different factors that hinder the application of centralized schemes. In first place, real systems may not have a model that capture correctly their dynamics. Moreover, even if a model can be obtained, it may be too complex to be useful to design a controller. Likewise, there are other important limitations. For example, the system may be geographically disperse, just as happens in transportation networks [1]. Other times it is a matter of privacy: the subsystems that compose the overall system may be independent and have incentives to keep some information secret. This could be, for example, the case of a supply chain. It is at this point where decentralized and distributed controllers come into play. The idea behind these schemes is simple: the centralized problem is divided in several different parts whose control is assigned to a certain number of local controllers or *agents*. These agents share information with the rest in order to improve closed-loop performance, robustness and fault-tolerance.

Decentralized and distributed systems have been a subject of study for a long time, but it has not been until the last decade when they have been at their very peak. The renewed interest in distributed and decentralized schemes has been mainly motivated by the proliferation of low cost wireless transceivers and their wide range of applications [2,3]. Wireless autonomous networks provide a mean to measure or actuate much cheaper than the traditional wired solutions.

There are several important issues that characterize distributed control problems. Aspects such as the way agents share information are crucial. The complexity of a distributed control problem roots in the fact that the performance of the closed-loop system depends on the decisions that all the agents take. Crossed interactions and side effects are key ideas to understand why cooperation and communication policies become very important issues. This problem is not new and has been studied by disciplines such as economics and game theory. Game theory is a theoretical framework that allows one to study the problem of cooperation of different agents with, maybe, conflicting control goals, from a mathematical point of view [4,5]. In an economic context the role played by undesired interactions between subsystems is studied under the idea of externalities [6].

In this work we focus on distributed model predictive control (MPC) schemes. MPC is a popular control strategy for the design of high performance model-based process control systems because of its ability to handle multi-variable interactions, constraints on control (manipulated) inputs and system states, and optimization requirements in a systematic manner. MPC takes advantage of a system model to predict its future evolution starting from the current system state along a given prediction horizon. The success of MPC in industrial applications [7] has motivated an important amount of research on the stability, robustness and optimality of model predictive controllers. Nevertheless, MPC has strong computational requirements which hinder its application to large-scale systems. Hence, it is natural to use a distributed approach for this class of systems. Several distributed MPC schemes have been proposed in the literature in the last years. Next, we provide a review of the most important contributions that can be found in the liter-

* Corresponding author.
E-mail address: davidmps@cartuja.us.es (D. Muñoz de la Peña).

0959-1524/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jprocont.2010.12.006

ature in this area. More comprehensive reviews of this topic can be found in [8] or [9]. In general, there are three different approaches to distribute the control problem among the agents. The first one is the mere decentralization of the problem. In [10] the centralized MPC problem is decentralized considering only a one step horizon, which guarantees small deviations in the values of the variables the agent share. In addition, a sufficient criterion for analyzing a posteriori the asymptotic stability of the process model in closed-loop with the set of decentralized MPC controllers is given. This work is enhanced in [11] for the case of packet loss. The second DMPC approach is based on information broadcast. In this category we include those DMPC schemes in which the agents communicate with the goal of providing useful information for the decisions of the rest of their neighbors. For example, in [12] a DMPC scheme for linear systems coupled only through the state is considered. In this scheme the agents exchange the predictions about their state at the end of each sample step. In [13] the DMPC controllers exchange bounds of their state trajectories and incorporate this information into their local problems. Other algorithms in the literature are based on an iterative procedure of information broadcast. For example, in [14] this procedure is presented as communication-based control. In [15] another iterative implementation of a similar DMPC scheme was applied together with a distributed Kalman filter to a quadruple tank system. Finally, the third approach is based on agent collaboration. In this category we include those DMPC schemes in which the agents exchange information trying to obtain a consensus on the values of the shared variables. For example, dual decomposition has been used for DMPC in [16]. An application for the control of irrigation canals can be found in [17]. In [18] a descent direction algorithm is used to control an urban traffic network. A different gradient-based distributed dynamic optimization method based on the exchange of sensitivities is proposed in [19]. In [20,21] a decentralized control architecture for nonlinear systems with continuous and asynchronous measurements was presented. Following up on this work, in [22,23] distributed model predictive control methods for the design of networked control systems based on Lyapunov-based model predictive control were presented. Finally, Jacobi algorithm [24] is the core idea of one of the feasible cooperation-based MPC used in [25–27].

In general, most of the previous works assume that the agents have full knowledge of the model and the state of the system, which can be to some extent restrictive. In this paper we assume that each agent has only local state and model information and propose a DMPC algorithm for multiple agents based on agent negotiation. This algorithm is a multiagent extension of the two agent scheme presented in [28]. Agents communicate in order to find a cooperative solution to the problem of controlling a set of constrained linear systems coupled through the inputs. At each sampling time, a negotiation takes place in which the agents make different proposals, from which only one of them is chosen following a social criterion. In addition, we provide sufficient conditions that guarantee practical stability of the closed-loop system along with an optimization based design procedure that is based on the full model of the system.

2. Problem formulation

We consider the following class of distributed linear systems in which there are M_x subsystems coupled with their neighbors through M_u inputs

$$x_i(t+1) = A_i x_i(t) + \sum_{j \in n_i} B_{ij} u_j(t) \tag{1}$$

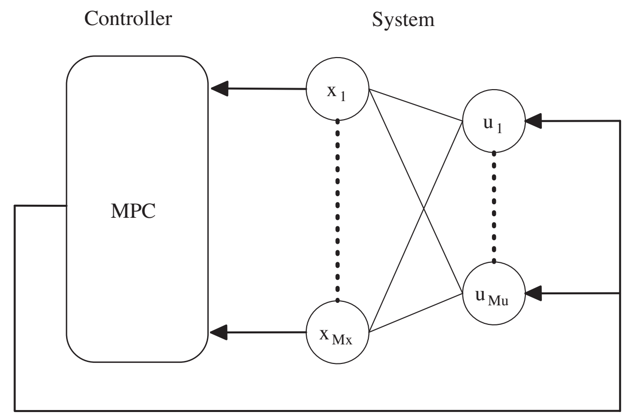

Fig. 1. Centralized MPC.

where $x_i \in R^{q_i}$ with $i = 1, \ldots, M_x$ are the states of each subsystem, and $u_j \in R^{r_j}$ with $j = 1, \ldots, M_u$ are the different inputs.[1] The set of indices n_i indicates the set of inputs u_j which affect the state x_i and the set of indices m_j indicates the set of states x_i affected by the input u_j. Note that this allow to define mathematically the concept of neighborhood of agent i as

$$N_i := \bigcup_{j \in n_i} m_j. \tag{2}$$

Therefore, any agent j included in N_i is a neighbor of agent i. Note that this does not imply that $i \in N_j$, that is, the neighborhood is not a symmetrical property in this context.

We consider the following linear constraints in the states and the inputs

$$\begin{aligned} x_i &\in \mathcal{X}_i, \quad i = 1, \ldots, M_x \\ u_j &\in \mathcal{U}_j, \quad j = 1, \ldots, M_u \end{aligned} \tag{3}$$

where $\mathcal{X}_i$ and $\mathcal{U}_j$ are closed polyhedra that contain the origin in their interior defined by the following set of linear inequalities.

$$\begin{aligned} x_i \in \mathcal{X}_i &\leftrightarrow H_{x_i} x_i \le b_{x_i}, \quad i = 1, \ldots, M_x \\ u_j \in \mathcal{U}_j &\leftrightarrow H_{u_j} u_j \le b_{u_j}, \quad j = 1, \ldots, M_u \end{aligned} \tag{4}$$

Note that, as these polyhedra contain the origin in their interior, then $b_{x_i} > 0$ and $b_{u_j} > 0$.

There are many different physical systems that can be modeled under this formulation. For example, in [18] this model is used to represent the dynamics of a traffic network. In [17] the dynamics of an irrigation canal system are described with a similar formulation. In [28] the beer game, a typical supply chain problem, is described likewise.

This class of systems can be represented by a graph in which to each node either the state of one of the subsystems or one of the inputs available is assigned, and the arcs connect the inputs to the states they affect.

The control objective is to regulate the states of all the subsystems to the origin while satisfying the state and input constraints. To this end, centralized MPC follows a receding horizon approach and at each sampling time obtains the current states and solves a single finite horizon optimal control problem based on a performance index that depends on all the states and inputs. See Fig. 1 for a scheme of a centralized MPC controller. In distributed MPC schemes there are several agents that decide all the control inputs. It can be seen that although the states are not dynamically coupled, the agents need to negotiate in order to decide the value of

[1] Throughout the paper the time dependence is omitted when possible for notational convenience.

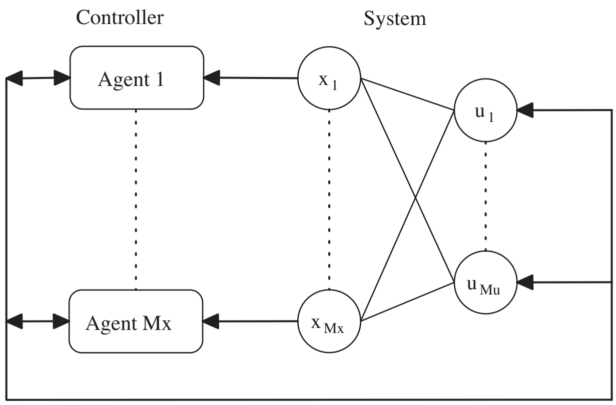

Fig. 2. Distributed MPC.

the shared inputs. There are many possible distributed schemes depending on the available information and communication constraints. Fig. 2 shows a scheme of a distributed controller in which each agent has access to partial state information and can communicate with the rest of the agents. This is the class of distributed control scheme considered in this work that is presented in the next section.

Remark 1. One of the differences between the proposed approach and other cooperative MPC schemes is that in order to implement the control law, the agents do not need to have a global model of the system. This may be important in some applications in which the centralized model is not available or the agents do not want to share this information with the rest of the subsystems. In addition, there is a potential benefit from this assumption because if a distributed system adds a new subsystem, in the proposed scheme, only those agents affected by this new element would have to be updated, while in other schemes based on global information, the information would have to be broadcasted. One class of systems in which these issues are relevant are transport networks and supply chains, where new consumers/suppliers can appear dynamically.

Remark 2. In the proposed scheme, several agents decide upon all or a subset of the control inputs. This implies that the inputs are not assigned to a particular agent as in most distributed MPC schemes found in the literature. Moreover, nothing is said about the magnitudes of M_x and M_u, thus this framework allows modeling situations in which there are agents with no associated inputs or even states. Hierarchical control or the existence of mediators in the network (agents that suggest an actuation for the rest of agents based on their own knowledge of the system) are examples of other interesting possibilities that can be also modeled with this framework.

3. Proposed DMPC controller

In this paper we propose a distributed scheme assuming that for each subsystem, there is an agent that has access to the model and the state of that subsystem. The agents do not have any knowledge of the dynamics of its neighbors, but can communicate freely among them in order to reach an agreement. The proposed strategy is based on negotiation between the agents. At each sampling time, following a given protocol, agents make proposals to improve an initial feasible solution on behalf of their local cost function, state and model. These proposals are accepted if the global cost improves the cost corresponding to the current solution. To this end, the agent that makes the proposal must communicate with the neighbors affected. Note that a proposal may modify only a subset of inputs, and hence there are agents that may not be affected by these changes. Different negotiation/communication protocols may be implemented. The only requirement is that the protocol must guarantee that each proposal is evaluated independently. In this paper, we propose to implement a controller in which at each sampling time, a fixed number of proposals made sequentially by random agents are considered.

The control objective of the proposed scheme is to minimize a global performance index defined as the sum of each of the local cost functions. The local cost function of agent i based on the predicted trajectories of its state and inputs defined as

$$J_i(x_i, \{U_j\}_{j\in n_i}) = \sum_{k=0}^{N-1} L_i(x_{i,k}, \{u_{j,k}\}_{j\in n_i}) + F_i(x_{i,N}) \tag{5}$$

where $U_j = \{u_{j,k}\}$ is the future trajectory of input j, N is the prediction horizon, $L_i(\cdot)$ with $i \in M_x$ is the stage cost function defined as

$$L_i(x_i, \{u_j\}_{j\in n_i}) = x_i^T Q_i x_i + \sum_{j\in n_i} u_j^T R_{ij} u_j \tag{6}$$

with $Q_i > 0$, $R_{ij} > 0$ and $F_i(\cdot)$ is the terminal cost defined as

$$F_i(x_i) = x_i^T P_i x_i \tag{7}$$

with $P_i > 0$. We use the notation $x_{i,k}$ to denote the state i, k-steps in the future obtained from the initial state x_i applying the input trajectories defined by $\{U_j\}_{j\in n_i}$. Note that each of the local cost functions only depends on the trajectories of its state and the inputs that affect it.

At the end of the negotiation rounds, the agents decide a set of input trajectories denoted as U^d. The first input of these trajectories is applied, however, the rest of the trajectories are not discarded, instead are used to generate the initial proposal for the next sampling round which is given by the shifted future input trajectories U^s of all the inputs. The last input of each of these trajectories is given by

$$\sum_{p\in m_j} K_{jp} x_{p,N} \tag{8}$$

where $x_{p,N}$ is the predicted values of the state x_p after N time steps obtained applying $U^d(t-1)$ from the initial state $x_p(t)$. The set of shifted input trajectories will be applied in case the agents do not reach an agreement. This proposal is necessary in order to guarantee closed-loop stability.

We define next the proposed distributed MPC scheme:

- Step 1: Each agent p measures its current state $x_p(t)$. The agents communicate in order to obtain $U^s(t)$ from $U^d(t-1)$. In order to do this, each agent must receive $K_{ji}x_{i,N}$ from each agent i such that $K_{ji} \neq 0$ for some $j \in n_p$. The initial value for the decision control vector $U^d(t)$ is set to the value of the shifted input trajectories, that is, $U^d(t) = U^s(t)$.
- Step 2: Randomly, agents try to submit their proposals. To this end, each agent asks the neighbors affected if they are free to evaluate a proposal (each agent can only evaluate a proposal at any given time). If all the agents acknowledge the petition, the algorithm continues. If not, the agent waits a random time before trying again. We will use the superscript p to refer to the agent which is granted permission to make a proposal.

- Step 3: In order to generate its proposal, agent p minimizes J_p solving the following optimization problem:

$$\begin{aligned}\{U_j^p(t)\}_{j\in n_p} = \ & arg \min_{\{U_j\}_{j\in n_p}} J_p(x_p, \{U_j\}_{j\in n_p}) \\ & s.t. \\ & x_{p,k+1} = A_p x_{p,k} + \sum_{j\in n_p} B_{pj} u_{j,k} \\ & x_{p,0} = x_i(t) \\ & x_{p,k} \in \mathcal{X}_p, \quad k = 0, \ldots, N \\ & u_{j,k} \in \mathcal{U}_j, \quad k = 0, \ldots, N-1, \quad \forall j \in n_p \\ & x_{p,N} \in \Omega_p \\ & U_j = U_j^d(t), \quad \forall j \notin n_{prop}\end{aligned} \tag{9}$$

  In this optimization problem, agent p optimizes over a set n_{prop} of inputs that affect its dynamics, that is, $n_{prop} \subseteq n_p$. Based on the optimal solution of this optimization problem, agent p presents a proposal defined by a set of input trajectories $\{U_j^p(t)\}_{j\in n_p}$ where $U_j^p(t)$ stands for the value of the trajectory of input j of the proposal of agent p. From the centralized point of view, the proposal at time step t of agent p is defined as

$$U^p(t) = \{U_j^p(t)\}_{j\in n_p} \uplus U^d(t) \tag{10}$$

  where the operation $\uplus$ stands for the update of the components relatives to $\{U_j^p(t)\}_{j\in n_p}$ in $U^d(t)$ and leaving the rest unmodified.
- Step 4: Each agent i who is affected by the proposal of agent p evaluates the predicted cost corresponding to proposed solution. To do so, the agent calculates the difference between the cost of the new proposal $U^p(t)$ and the cost of the current accepted proposal $U^d(t)$ as

$$\Delta J_i^p(t) = J_i(x_i(t), \{U_j^p(t)\}_{j\in n_i}) - J_i(x_i(t), \{U_j^d(t)\}_{j\in n_i}) \tag{11}$$

  This difference $\Delta J_i^p(t)$ is sent back to the agent p. If the proposal does not satisfy the constraints of the corresponding local optimization problem, an infinite cost increment is assigned. This implies that unfeasible proposals will never be chosen.
- Step 5: Once agent p receives the local cost increments from each neighbor, it can evaluate the impact of its proposal $\Delta J^p(t)$, which is given by the following expression

$$\Delta J^p(t) = \sum_{i \in \bigcup_{j\in n_{prop}} m_j} \Delta J_i^p(t) \tag{12}$$

  This global cost increment is used to make a cooperative decision on the future inputs trajectories. If $\Delta J^p(t)$ is negative, the agent will broadcast the update on the control actions involved in the proposal and the joint decision vector $U^d(t)$ will be updated to the value of $U^p(t)$, that is $U^d(t) = U^p(t)$. Else, is discarded.
- Step 6: The algorithm goes back to step 1 until the maximum number of proposals have been made or the sampling time ends. We denote the cost corresponding to the decided inputs as

$$J(t) = \sum_{i=1}^{M_x} J_i(x_i(t), \{U_j^d(t)\}_{j\in n_i}) \tag{13}$$

- Step 7: The first input of each sequence in $U^d(t)$ is applied and the procedure is repeated the next sampling time.

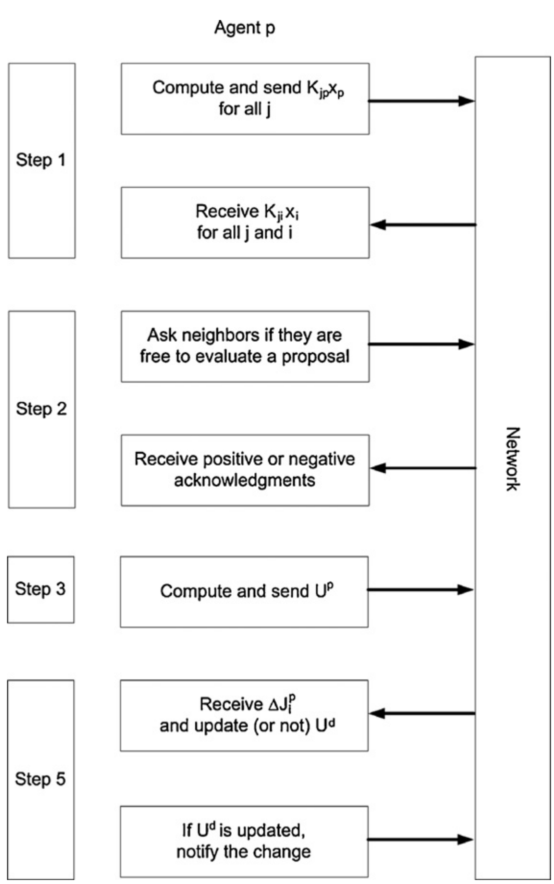

Fig. 3. Flow diagram for a single agent which is granted permission to make a proposal of the proposed DMPC scheme.

In Fig. 3 a flow diagram for a single agent of the proposed DMPC scheme is shown assuming that all the states are affected by all the inputs (hence, all the agents are neighbors). It can be seen that the agent must communicate several times with the rest of the agents. Note that in order to implement the proposed algorithm, it is necessary to obtain a set of future input trajectories that satisfy all the constraints for the initial state; that is, to initialize $U^s(0)$.

From a game theory point of view the situation can be described as a cooperative team game in which the possible strategies for each player are defined by its own proposals and the proposals of the rest of the agents. The utility of the proposals for each agent is defined by its local cost function, however, in order to find a solution, each agent chooses the option that is best from the global point of view.

Remark 3. The time variable t, which is always used between parenthesis, references sampling times. The variable k, which is used always as a subscript, references the future time steps along the prediction horizon of a given optimization problem and always takes values between 0 and N.

Remark 4. Several proposals can be evaluated in parallel as long as they do not involve the same set of agents; that is, at any given time an agent can only evaluate a single proposal. The communication protocol to implement the algorithm in parallel is beyond the scope of this work.

Remark 5. Centralized MPC solves a single large-scale problem based on the model of the whole system such as the following optimization problem:

$$
\begin{aligned}
\{U_j^c\}_{j=1,\ldots,M_u} = \;& \arg\min_{\{U_j\}_{j=1,\ldots,M_u}} \sum_{i=1}^{M_x} J_i(x_i, \{U_j\}_{j\in n_i}) \\
& s.t. \\
& x_{i,k+1} = A_i x_{i,k} + \sum_{j\in n_i} B_{ij} u_{j,k} \\
& x_{i,0} = x_i \\
& x_{i,k} \in \mathcal{X}_i, \quad k = 0,\ldots,N \\
& u_{j,k} \in \mathcal{U}_j, \quad k = 0,\ldots N-1, \quad \forall j \in n_i \\
& x_{i,N} \in \Omega_i \\
& \forall i = 1,\ldots,M_x
\end{aligned}
\tag{14}
$$

4. Stability

Stability is a major issue in distributed systems. In general, it is a difficult problem because it is not enough to guarantee the stability of each of the subsystems. Actually, stable subsystems may lead to an unstable global system. In this section we provide sufficient conditions that guarantee asymptotic stability of the closed-loop system following a standard region/terminal cost approach [29].

Assumption 1. There exist linear feedback $u_j = \sum_{p\in m_j} K_{jp} x_p$ and sets $\Omega_i \subseteq R^{q_i}$ such that if $x_i \in \Omega_i$ for all $i = 1, \ldots, M_x$ then the following conditions hold for all $i = 1, \ldots, M_x$

$$\sum_{i=1}^{M_x} F_i(A_i x_i + \sum_{j\in n_i} B_{ij} \sum_{p\in m_j} K_{jp} x_p) - F_i(x_i) + L_i\left(x_i, \left\{\sum_{p\in m_j} K_{jp} x_p\right\}_{j\in n_i}\right) \le 0 \tag{15a}$$

$$A_i x_i + \sum_{j\in n_i} B_{ij} \sum_{p\in m_j} K_{jp} x_p \in \Omega_i \tag{15b}$$

$$\sum_{p\in m_j} K_{jp} x_p \in U_j \tag{15c}$$

$$\Omega_i \in X_i \tag{15d}$$

The requirements of Assumption 1 are twofold, first, the local feedbacks must satisfy constraint (15a) which implies that the system in closed-loop with these set of local controllers is stable. Second, sets Ω_i such that (15b)–(15d) are satisfied must exist. We denote these sets as jointly positive invariants for system (1) in closed-loop with the controllers defined by matrices K_{ij}. It is important to note that although the cartesian product of these sets is a positive invariant of system (1), in general it is not possible to obtain the jointly positive invariant sets from an invariant set of system (1) obtained following standard procedures because each Ω_i is defined only in a subspace of the whole state space; that is, in the space corresponding to the state x_i. This property is necessary in order to define for each agent a set of constraints that depend only on its state, and hence, only on its model. See the constraints of problem (9).

Theorem 1. *If Assumption 1 holds and at time step $t = 0$, $U^s(0)$ is given such that each of the M_x optimization problems (9)[2] are feasible for $x_{i,0} = x_i(0)$ and $U_j = U_j^s(0)$ with $i = 1, \ldots, M_x$ and $j \in n_i$, then the proposed algorithm is feasible for all time steps $t \ge 0$ and system (1) in closed-loop with the proposed distributed MPC controller is asymptotically stable.*

Proof. The proof consists of two parts. We first prove that there is always a proposal which satisfies all the constraints (9) and then we prove that, under the stated assumptions,

$$J(t) = \sum_{i=1}^{M_x} J_i(x_i(t), \{U_j^d(t)\}_{j\in n_i}) \tag{16}$$

is decreasing sequence lower-bounded by zero.

Part 1. Taking into account that $\{U_j^d(t-1)\}_{j\in n_i}$ satisfies all the constraints of (9) and Assumption 1, it is easy to prove that $\{U_j^s(t)\}_{j\in n_i}$ provides a feasible solution for $x_i(t)$. It follows, that $U^d(t)$ provides a feasible solution for the optimization problem of agent i because it is chosen among a set of proposals which are required to be feasible in order to be accepted. Note that a proposal which is unfeasible for any of the agents cannot be chosen because the corresponding local cost is infinite. Taking into account that by assumption, $U^s(0)$ satisfies all the constraints for all the agents at time step $t = 0$ and using the above result recursively, the statement of this part is proved.

Part 2. Taking into account the definitions of $U_i^d(t-1)$ and $U_i^s(t)$ it follows that

$$J_i(x_i(t), \{U_j^s(t)\}_{j\in n_i}) - J_i(x_i(t-1), \{U_j^d(t-1)\}_{j\in n_i}) \tag{17}$$

is equal to

$$
\begin{aligned}
&F_i\left(A_i x_{i,N} + \sum_{j\in n_i} B_{ij} \sum_{p\in m_j} K_{jp} x_{p,N}\right) - F_i(x_{i,N}) \\
&+ L_i\left(x_{i,N}, \left\{\sum_{p\in m_j} K_{jp} x_{p,N}\right\}_{j\in n_i}\right) - L_i(x_{i,0}, \{u_{j,0}\}_{j\in n_i})
\end{aligned}
\tag{18}
$$

where $u_{j,0}$ is the first input of $U_j^d(t-1)$. Taking into account (15a), this implies that

$$\sum_{i=1}^{M_x} J_i(x_i(t), \{U_j^s(t)\}_{j\in n_i}) - J(t-1) \le -\sum_{i=1}^{M_x} L_i(x_{i,0}, \{u_{j,0}\}_{j\in n_i}) \tag{19}$$

As the proposed algorithm chooses $U^d(t)$ as an input trajectory that improves the cost, it is easy to see that

$$J(t) \le J(t-1) - \sum_{i=1}^{M_x} L_i(x_{i,0}, \{u_{j,0}\}_{j\in n_i}) \tag{20}$$

Taking into account that recursive feasibility is guaranteed (see the first part of the proof) and the definitions of F_i and L_i and following the same lines of though as in [29] or [30], attractiveness and stability can also be proved. This implies that system (1) in closed-loop with the proposed distributed MPC controller is asymptotically stable. □

The proof of Theorem 1 follows the standard terminal region/terminal constraint approach, see [29]. Stability is inherited from the set of local controllers defined by matrices K_{ij} which by (15a) are known to stabilize the system. In fact this result is based on the well known idea "Feasibility implies stability", see [31].

Remark 6. The stability properties of the proposed scheme rely heavily on the fact that U_s satisfies all the constraints of the optimization problem. This implies, that in the start-up and when the controller loses feasibility due to disturbances, U_s has to be calculated either by a centralized supervisor or in a distributed manner by the agents.

[2] Although we used the index p in the definition of the optimization problems solved to obtain each proposal, in the proof of Theorem 1 we will use the index i.

Remark 7. When applied to a real system in the presence of disturbances and/or possible model errors, if the controller operates close to the state constraints in practice the shifted input trajectory may become unfeasible and it would have to be evaluated again (in a centralized manner or using an appropriate distributed approach). This issue must be taken into account in the implementation procedure of this control strategy.

5. Controller design procedure

The local controllers K_{ij} must satisfy two necessary conditions. First, the centralized system composed by the M_x subsystems (1) in closed-loop with the local controllers must be stable. Second, the jointly invariant sets must exist.

The local controllers that depend on each agent must be designed in a way such that (15a) holds. To take this condition into account, we will use the following centralized model of the system

$$x(t+1) = Ax(t) + Bu(t) \tag{21}$$

where

$$x = [x_1^T, \ldots, x_{M_x}]^T, \quad u = [u_1^T, \ldots, u_{M_u}]^T \tag{22}$$

and matrices A and B are appropriate matrices that depend of the model (1) of each subsystem.

In addition, stability of each subsystem in closed-loop with its corresponding local feedback must be guaranteed. A sufficient condition to guarantee stability of each of the subsystems is to require that the cost function defined by the matrices P_i is a Lyapunov function for the subsystem in closed-loop with its corresponding local feedback. To take into account this condition, we will use the following uncertain model of each of the M_x subsystems

$$x_i(t+1) = A_i x_i(t) + B_i v_i(t) + E_i w_i(t) \tag{23}$$

where $v_i \in R^{\sum_{j\in n_i} r_j}$ is made of the part of the inputs that affect x_i and depend on x_i and $w_i \in R^{\sum_{j\in n_i} r_j}$ is the part of the inputs that affect x_i depend on the rest of the states when the local controllers are applied; that is,

$$\begin{aligned} v_i(t) &= \{K_{ji}x_i(t)\}_{j\in n_i} \\ w_i(t) &= \left\{ \sum_{p\in m_j-\{i\}} K_{jp}x_p(t) \right\}_{j\in n_i} \end{aligned} \tag{24}$$

In this case, the objective is to design a controller $K_i = \{K_{ji}\}_{j\in n_i}$ that stabilizes the subsystem considering w_i an unknown disturbance. Matrices B_i and E_i are equal and depend of the model (1) of each subsystem, in particular

$$B_i = E_i = \{B_{ij}\}_{j\in n_i}$$

We provide next a set of linear matrix inequalities (LMI) that guarantees that (15a) holds and that K_i stabilizes the subsystem i. These LMI constraints are obtained following standard procedures, see for example [32–34].

Theorem 2. *Consider system (1). If there exist matrices W_i, Y_i with $i = 1, \ldots, M_x$ such that the following inequalities hold* [3]

$$\begin{bmatrix} \Upsilon & \Phi & \Psi & \Xi \\ * & \Upsilon & 0 & 0 \\ * & * & I & 0 \\ * & * & * & I \end{bmatrix} \geq 0 \tag{25}$$

with $R_i = \sum_{j\in n_i} R_{ij}$, $R = diag(R_1, \ldots, R_{M_x})$, $K = [K_1, \ldots, K_{M_x}]$, $K_i^T = [K_{1i}, \ldots, K_{M_u i}]$ and

$$\Phi = \begin{bmatrix} W_1A_1^T + Y_1^TB_1^T & Y_1^TB_2^T & \cdots & Y_1^TB_{M_x}^T \\ Y_2^TB_1^T & W_2A_2^T + Y_2^TB_2^T & \cdots & Y_2^TB_{M_x}^T \\ \vdots & \vdots & \ddots & \vdots \\ Y_{M_x}^TB_1^T & Y_{M_x}^TB_2^T & \cdots & W_{M_x}A_{M_x}^T + Y_{M_x}^TB_{M_x}^T \end{bmatrix} \tag{26}$$

$$\Upsilon = \begin{bmatrix} W_1 & 0 & \cdots & 0 \\ * & W_2 & \cdots & 0 \\ * & * & \ddots & \vdots \\ * & * & * & W_{M_x} \end{bmatrix}, \quad \Xi = \begin{bmatrix} Y_1^TR^{1/2} \\ Y_2^TR^{1/2} \\ \vdots \\ Y_{M_x}^TR^{1/2} \end{bmatrix},$$

$$\Psi = \begin{bmatrix} W_1Q_1^{1/2} & 0 & \cdots & 0 \\ * & W_2Q_2^{1/2} & \cdots & 0 \\ * & * & \ddots & \vdots \\ * & * & * & W_{M_x}Q_{M_x}^{1/2} \end{bmatrix}, \tag{27}$$

and

$$\begin{bmatrix} W_i & W_iA_i^T - Y_i^TB_i^T & W_iQ_i^{1/2} & Y_i^TR_i^{1/2} \\ * & W_i & 0 & 0 \\ * & * & I & 0 \\ * & * & * & I \end{bmatrix} \geq 0 \tag{28}$$

for $i = 1, \ldots, M_x$ then (15a) is satisfied for the matrices $P_i = W_i^{-1}$, $K_i = \{K_{ji}\}_{j\in n_i} = Y_iW_i^{-1}$ and systems (23) are stable in closed-loop with $v_i = K_ix_i$.

Proof. We will prove the theorem in two parts. In the first part we will prove that if (25) holds, then (15a) is satisfied for the matrices $P_i = W_i^{-1}$, $K_i = \{K_{ji}\}_{j\in n_i} = Y_iW_i^{-1}$. In the second part, we will prove that if (28) holds then system (23) is stable in closed-loop with $v_i = K_ix_i$.

Part 1: In this part, we will prove that (25) is equivalent to (15a). Taking into account the definition of the centralized system (21), (15a) can be posed as follows

$$(A+BK)^TP(A+BK) - P + Q + K^TRK \leq 0 \tag{29}$$

with

$$\begin{aligned} R &= \mathrm{diag}(R_1, \ldots, R_{M_x}) \\ Q &= \mathrm{diag}(Q_1, \ldots, Q_{M_x}) \\ P &= \mathrm{diag}(P_1, \ldots, P_{M_x}) \end{aligned} \tag{30}$$

with $R_i = \sum_{j\in n_i} R_{ij}$. Taking into account that the P and P^{-1} are positive defined matrices, if we multiply (29) by minus one and apply the Schur's complement we can recast (29) as the following constraint

$$\begin{bmatrix} P - Q - K^TRK & (A+BK)^T \\ (A+BK) & P^{-1} \end{bmatrix} \geq 0 \tag{31}$$

This LMI can be transformed into an equivalent one by pre and post multiplying it by a positive definite matrix

$$\begin{bmatrix} P^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} P - Q - K^TRK & (A+BK)^T \\ (A+BK) & P^{-1} \end{bmatrix} \begin{bmatrix} P^{-1} & 0 \\ 0 & I \end{bmatrix} \geq 0 \tag{32}$$

The resulting equivalent matrix inequality is given by

$$\begin{bmatrix} P^{-1} - P^{-1}QP^{-1} - P^{-1}K^TRKP^{-1} & P^{-1}(A+BK)^T \\ (A+BK)P^{-1} & P^{-1} \end{bmatrix} \geq 0 \tag{33}$$

[3] The symbol "$*$" stands for the symmetric part of a matrix.

In order to obtain a LMI inequality let $\Upsilon = P^{-1} = \text{diag}(W_1, W_2, \ldots, W_{M_x})$ with $W_i = P_i^{-1}$ for $i = 1, 2, .., M_x$ and $Y = K\Upsilon = [Y_1 \ Y_2 \ \ldots \ Y_{M_x}]$. It follows that

$$\begin{bmatrix} \Upsilon - \Upsilon Q \Upsilon - Y^T R Y & \Upsilon A^T + Y^T B^T \\ A\Upsilon + BY & \Upsilon \end{bmatrix} \geq 0 \tag{34}$$

Using the decomposition $Q = Q^{1/2}Q^{1/2}$ and applying Schur's complement we obtain

$$\begin{bmatrix} \Upsilon - Y^T R Y & \Upsilon A^T + Y^T B^T \\ A\Upsilon + BY & \Upsilon \end{bmatrix} - \begin{bmatrix} \Upsilon Q^{1/2} \\ 0 \end{bmatrix} I \begin{bmatrix} Q^{1/2}\Upsilon & 0 \end{bmatrix} \geq 0 \tag{35}$$

$$\begin{bmatrix} \Upsilon - Y^T R Y & \Upsilon A^T + Y^T B^T & \Upsilon Q^{1/2} \\ A\Upsilon + BY & \Upsilon & 0 \\ Q^{1/2}\Upsilon & 0 & I \end{bmatrix} \geq 0 \tag{36}$$

The same procedure is repeated for $R = R^{1/2}R^{1/2}$ obtaining

$$\begin{bmatrix} \Upsilon & \Upsilon A^T + Y^T B^T & \Upsilon Q^{1/2} \\ A\Upsilon + BY & \Upsilon & 0 \\ Q^{1/2}\Upsilon & 0 & I \end{bmatrix} - \begin{bmatrix} Y^T R^{1/2} \\ 0 \\ 0 \end{bmatrix} I \begin{bmatrix} R^{1/2} Y & 0 & 0 \end{bmatrix} \geq 0 \tag{37}$$

$$\begin{bmatrix} \Upsilon & \Upsilon A^T + Y^T B^T & \Upsilon Q^{1/2} & Y^T R^{1/2} \\ A\Upsilon + BY & \Upsilon & 0 & 0 \\ Q^{1/2}\Upsilon & 0 & I & 0 \\ R^{1/2} Y & 0 & 0 & I \end{bmatrix} \geq 0 \tag{38}$$

Defining $\Phi = \Upsilon A^T + Y^T B^T$, $\Psi = \Upsilon Q^{1/2}$ and $\Xi = Y^T R^{1/2}$, the following LMI constraint is obtained and hence the proof is completed

$$\begin{bmatrix} \Upsilon & \Phi & \Psi & \Xi \\ * & \Upsilon & 0 & 0 \\ * & * & I & 0 \\ * & * & * & I \end{bmatrix} \geq 0 \tag{39}$$

Part 2: In this part, we will prove that if (28) holds then system (23) is stable in closed-loop with $v_i = K_i x_i$. To this end, we will prove that (28) is equivalent to the following constraint

$$(A_i + B_i K_i)^T P_i (A_i + B_i K_i) - P_i + Q_i + K_i^T R_i K_i \leq 0 \tag{40}$$

which implies that $V_i(x) = x_i^T P_i x_i$ is a Lyapunov function of the closed-loop system and hence is stable. To prove this part of the theorem the constraint (40) is transformed in its equivalent LMI constraint (28) following the same procedure used in the first part. □

Remark 8. Additional constraints can be added to the design procedure so that there is no need to know the state x_i in order to calculate the input u_j. This is relevant because in order to evaluate the shifted input trajectory, all the subsystems whose state affects a given input must communicate, so in certain cases, it may be desirable to limit these communications.

Once the local controllers and the terminal cost functions are fixed, in order to design a distributed MPC scheme that satisfies the assumptions of Theorem 1 one needs to find sets Ω_i such that (15b)–(15d) hold. In general this is a difficult problem because each of the sets depends on the others. The size of the terminal region for agent i is determined by the magnitude of the disturbances induced by its neighbor agents and viceversa. A similar class of invariant systems was studied in [35] within the polytopic games framework. We provide next an optimization based procedure to solve this problem. In order to present the algorithm we need the following definitions.

Definition 1. Given the following discrete-time linear system subject to bounded additive uncertainties

$$x^+ = \hat{A}x + \hat{B}u + \hat{E}w \tag{41}$$

with $w \in \hat{\mathcal{W}}$, subject to constraints in the state and the input $x \in \hat{\mathcal{X}}$, $u \in \hat{\mathcal{U}}$ and a linear feedback $u = \hat{K}x$; a set Ω is said to be a robust positive invariant set for the system if the following constraints hold

$$\begin{aligned} & x \in \Omega \rightarrow (\hat{A} + \hat{B}\hat{K})x + \hat{E}w \in \Omega, \quad \forall w \in \hat{\mathcal{W}} \\ & \hat{K}x \in \hat{\mathcal{U}} \\ & \Omega \subseteq \hat{\mathcal{X}} \end{aligned} \tag{42}$$

Given system matrices $\hat{A}, \hat{B}, \hat{E}, \hat{K}$ and the sets $\hat{\mathcal{X}}, \hat{\mathcal{U}}, \hat{\mathcal{W}}$, there exists several methods to find a set Ω that satisfies these constraints, see for example [36] for a procedure to find the maximal robust positive invariant and [37] for a procedure to find an approximation of the minimal robust positive invariant. We denote $\Omega(\hat{A}, \hat{B}, \hat{E}, \hat{\mathcal{X}}, \hat{K}, \hat{\mathcal{U}}, \hat{W})$ the corresponding maximal robust positive invariant set.

In order to obtain sets Ω_i such that Assumption 1 is satisfied, we will use the uncertain model (23) of each agent; that is, each agent assumes that the contribution of its neighbors to the inputs that affect its dynamics are an unknown bounded disturbance. The size of the set in which these disturbances are bounded depend on the size of the sets Ω_i. This implies that finding these sets is in general a complex problem. In order to decouple the design of each set, each agent i limits its contribution to each input j by a factor $\lambda_{ji} \in (0, 1]$ with $\sum_{i \in m_j} \lambda_{ji} \leq 1$; that is,

$$K_{ji} x_i \in \lambda_{ji} \mathcal{U}_j, \quad \forall i, j \tag{43}$$

Using the same notation introduced in (23), this implies that

$$v_i \in \mathcal{V}_i(\Lambda), \quad w_i \in \mathcal{W}_i(\Lambda) \tag{44}$$

with

$$\begin{aligned} & V_i(\Lambda) = \lambda_{1i}\mathcal{U}_1 \times \ldots \times \lambda_{M_u i}\mathcal{U}_{M_u} \\ & W_i(\Lambda) = \Big(\sum_{p \in m_1 - \{i\}} \lambda_{1p}\Big)\mathcal{U}_1 \times \ldots \times \Big(\sum_{p \in m_{M_u} - \{i\}} \lambda_{M_u p}\Big)\mathcal{U}_{M_u} \end{aligned} \tag{45}$$

where $\Lambda = \{\lambda_{ij}\}_{\forall i,j}$ is a vector made of all the parameters λ_{ij}. Note that the maximum contribution of a given agent inside Ω_i is the maximum contribution to the disturbance for the rest of the agents. In order to decouple the computation of the jointly invariant sets Ω_i, we use the following result based on finding a robust positive invariant set for each subsystem:

Lemma 1. *Given constants $\lambda_{ji} \in (0, 1)$ with $\sum_{i \in m_j} \lambda_{ji} \leq 1$, if the sets defined as*

$$\Omega_i = \Omega(A_i, B_i, E_i, \mathcal{X}_i, K_i, \mathcal{V}_i(\Lambda), \mathcal{W}_i(\Lambda)) \tag{46}$$

are not empty, they satisfy the constraints (15b)–(15d).

The lemma stems from the definition of the operator Ω. If all the sets exist, then they satisfy the stability constraints. Note that there exists an infinite number of possible values of λ_{ji} such that these sets exist. In order to chose one, we propose to solve the following optimization problem which maximizes the feasibility region of the distributed MPC controller:

$$\begin{aligned} \max_{\lambda_{ji}} \quad & f(\Omega_1 \times \Omega_2 \ldots \times \Omega_{M_x}) \\ & \Omega_i = \Omega(A_i, B_i, E_i, \mathcal{X}_i, K_i, \mathcal{V}_i(\Lambda), \mathcal{W}_i(\Lambda)) \\ & \lambda_{ji} \in (0, 1), \quad \forall j, i \\ & \sum_{i \in m_j} \lambda_{ji} \leq 1, \quad \forall j \end{aligned} \tag{47}$$

where function $f(\cdot)$ is a measure of the size of a polyhedron (for example, its Chebyshev radius).

Solving problem (47) may be difficult in general, however, under certain assumptions it can be posed as a convex problem. In [35] it was proved that the feasibility region of this problem is convex. In the next lemma we prove that the jointly invariant sets Ω_i are polyhedra defined by a set of inequalities whose right hand side can be expressed as an affine combination of the constants λ_{ij}. This implies, that if an appropriate function $f(\cdot)$ is chosen, problem (47) can be cast into a convex optimization problem.

Lemma 2. *If $A_i+\sum_j B_{ij}K_{ji}$ is stable, then the set*

$$\Omega_i=\Omega(A_i,B_i,E_i,\mathcal{X}_i,K_i,\mathcal{V}_i(\Lambda),\mathcal{W}_i(\Lambda)) \tag{48}$$

is a polyhedron that can be defined as a set of inequalities whose independent term can be expressed as an affine combination of the constants λ_{ij}, that is,

$$\Omega_i=\{x_i: M_i x_i \le b_i+\sum_{j\in n_i}\sum_{p\in m_j}\lambda_{jp}b_{ij}\} \tag{49}$$

Proof. The calculation of the robust invariant for a linear system is a well known problem and several procedures can be found in the literature, for instance in [38] or [36]. In order to prove the lemma, we will follow the procedure presented in [36]. The main idea is to find the set of states such that the trajectories of the closed-loop system starting from these states fulfill all the state and input constraints for all possible disturbances. This is done in an iterative manner. The set of states that fulfill the constraints after k steps is determined for increasing values of k. This process is repeated until convergence is obtained, that is, the same set of states is obtained for k and $k+1$. The resulting set is the maximum invariant set. Note that each value of k adds new constraints that the invariant set must fulfill, so the number of restrictions grows with each iteration. First of all, we will define the state constraints that the closed-loop system has to satisfy taking into account the constraints in u_{ji}, the contribution of the state x_i to the different inputs u_j. By definition of Ω_i, u_{ji} has to verify

$$u_{ji}=K_{ji}x_i\in\lambda_{ji}\mathcal{U}_j,\quad j\in n_i \tag{50}$$

Hence, the input constraint condition for the input j (4) can be transformed into the following set of inequalities:

$$H_{u_j}K_{ji}x_i\le\lambda_{ji}b_{u_j},\quad j\in n_i \tag{51}$$

Note that as $\lambda_{ji}\in(0,1)$, the set of inequalities is equal or more restrictive than the original input constraints. These inequalities have to be taken into account in the state constraints of the closed-loop system. The new set of state constraints can be written as

$$\hat{H}_{x_i}x_i\le\hat{b}_{x_i} \tag{52}$$

For example, if $n_i=\{1,2,\ldots,M_u\}$, that is, subsystem i is affected by all the inputs, then

$$\hat{H}_{x_i}=\begin{bmatrix} H_{x_i}\\ H_{u_1}K_{1i}\\ \vdots\\ H_{u_{M_u}}K_{M_u i}\end{bmatrix},\quad \hat{b}_{x_i}=\begin{bmatrix} b_{x_i}\\ \lambda_{1i}b_{u_1}\\ \vdots\\ \lambda_{M_u i}b_{u_{M_u}}\end{bmatrix} \tag{53}$$

Note that the right hand side of the inequalities can be expressed as an affine combination of the constants λ_{ij} with $j\in n_i$. Let $A_{CL_i}=(A_i+\sum_j B_{ij}K_{ji})$. If A_{CL_i} is stable, then for each value of Λ, the robust invariant set Ω_i can be determined in a finite number of steps forward $k(\Lambda)$. Let $k^*=\max_{\Lambda}k(\Lambda)$. We can compute the robust invariant set for all possible values of Λ as the set of states such that its k-steps ahead predictions satisfy all the constraints for all possible future disturbances; that is,

$$\hat{H}_{x_i}\left(A_{CL_i}^k x_i+\sum_{g=0}^{k-1}A_{CL_i}^g\sum_{j\in n_i}\sum_{p\in m_j-\{i\}}B_{ij}u_{jp}\right)\le\hat{b}_{x_i},\quad k=1,\ldots,k^* \tag{54}$$

for all $u_{jp}\in\lambda_{jp}\mathcal{U}_j$ with $j\in n_i$ and $p\in m_j-\{i\}$. Taking into account that for all $u_{jp}\in\lambda_{jp}\mathcal{U}_j$ with $p\in m_j-\{i\}$ there exists $z_j\in\mathcal{U}_j$ such that

$$z_j\sum_{p\in m_j-\{i\}}\lambda_{jp}=\sum_{p\in m_j-\{i\}}u_{jp} \tag{55}$$

constraint (54) is equivalent to

$$\hat{H}_{x_i}(A_{CL_i}^k x_i+\sum_{g=0}^{k-1}A_{CL_i}^g\sum_{j\in n_i}B_{ij}z_j\sum_{p\in m_j-\{i\}}\lambda_{jp})\le\hat{b}_{x_i},\quad k=1,\ldots,k^* \tag{56}$$

for all $z_j\in\mathcal{U}_j$ with $j\in n_i$. In order to eliminate the disturbance from the constraints and obtain a deterministic set, let us focus on each of the n_r rows of $\hat{H}_{x_i}$ (which define a different constraint for each time step k taken into account in the definition of the invariant set). To denote the r-th row of a matrix A we will use the $[A]_r$. Using this notation, constraint (56) is equivalent to

$$\begin{aligned}&[\hat{H}_{x_i}A_{CL_i}^k]_r x_i\le[\hat{b}_{x_i}]_r-\left[\hat{H}_{x_i}\sum_{g=0}^{k-1}A_{CL_i}^g\sum_{j\in n_i}B_{ij}z_j\sum_{p\in m_j-\{i\}}\lambda_{jp}\right]_r,\\ &k=1,\ldots,k^*,\quad r=1,\ldots,n_r\end{aligned} \tag{57}$$

Let us define

$$\sigma_{ij}^{gr}=\max_{z_j\in\mathcal{U}_j}([(\hat{H}_{x_i}A_{CL_i}^g B_{ij})]_r z_j) \tag{58}$$

Note that σ_{ij}^{gr} is a scalar that can be calculated from the system model and constraints. This definition allows us to rewrite constraint (57) as:

$$\begin{aligned}&[\hat{H}_{x_i}A_{CL_i}^k]_r x_i\le[\hat{b}_{x_i}]_r-\sum_{g=0}^{k-1}\sum_{j\in n_i}\sigma_{ij}^{gr}\left(\sum_{p\in m_j-\{i\}}\lambda_{jp}\right),\\ &\quad k=1,\ldots,k^*,\ \ r=1,\ldots,n_r\end{aligned} \tag{59}$$

Taking into account that the second term of each of the constraints of (59) is an affine combination of the constants $\{\lambda_{ip}\}$ it is possible to find matrix M_i and vectors b_i and b_{ij} with $j\in n_i$ such that

$$\Omega_i=\{x_i: M_i x_i\le b_i+\sum_{j\in n_i}\sum_{p\in m_j}\lambda_{jp}b_{ij}\} \tag{60}$$

□

Using this result, the problem of finding a matrix Λ that maximizes a measure of the distance can be cast into a convex optimization problem. For instance, let us suppose that our criterium to compare the invariant sets is the radium of a Chebyshev ball inside the invariant region. In this case we are interested in obtaining the maximum $x^T x$ as function of Λ that verifies all the constraints, which is a convex problem.

Remark 9. Although in order to implement the proposed controller, the agents do not need information about the state or the dynamics of the rest of the subsystems, a centralized model of the full system is needed in order to design the terminal region and the terminal constraint using the proposed design method so that closed-loop stability is guaranteed. Note that it is not mandatory to design the controller to satisfy the stability assumptions in order to implement the proposed distributed strategy.

6. Example

Consider a system of the form (1) defined by the following matrices

$$A_1=\begin{bmatrix}1 & 0.8\\0 & 0.7\end{bmatrix},\quad B_{11}=\begin{bmatrix}0\\1\end{bmatrix},\quad B_{12}=\begin{bmatrix}0\\0.15\end{bmatrix},\quad B_{13}=\begin{bmatrix}0\\0.15\end{bmatrix},\quad B_{14}=\begin{bmatrix}0\\0\end{bmatrix}$$
$$A_2=\begin{bmatrix}1 & 0.6\\0 & 0.7\end{bmatrix},\quad B_{21}=\begin{bmatrix}0\\0.15\end{bmatrix},\quad B_{22}=\begin{bmatrix}0\\1\end{bmatrix},\quad B_{23}=\begin{bmatrix}0\\0\end{bmatrix},\quad B_{24}=\begin{bmatrix}0\\0.15\end{bmatrix}$$
$$A_3=\begin{bmatrix}1 & 0.9\\0 & 0.8\end{bmatrix},\quad B_{31}=\begin{bmatrix}0\\0.15\end{bmatrix},\quad B_{32}=\begin{bmatrix}0\\0\end{bmatrix},\quad B_{33}=\begin{bmatrix}0\\1\end{bmatrix},\quad B_{34}=\begin{bmatrix}0\\0.15\end{bmatrix}$$
$$A_4=\begin{bmatrix}1 & 0.8\\0 & 0.5\end{bmatrix},\quad B_{41}=\begin{bmatrix}0\\0\end{bmatrix},\quad B_{42}=\begin{bmatrix}0\\0.15\end{bmatrix},\quad B_{43}=\begin{bmatrix}0\\0.15\end{bmatrix},\quad B_{44}=\begin{bmatrix}0\\1\end{bmatrix} \tag{61}$$

subject to the following linear constraints in the state and the inputs

$$\begin{aligned}&|x_1|_\infty\le 1,\quad |x_2|_\infty\le 2,\quad |x_3|_\infty\le 1,\quad |x_4|_\infty\le 2\\&|u_1|_\infty\le 1,\quad |u_2|_\infty\le 1,\quad |u_3|_\infty\le 1,\quad |u_4|_\infty\le 1\end{aligned} \tag{62}$$

A graph that represents the couplings between the individual subsystems can be seen in Fig. 4. The boxes represent the subsystems while the arrows represent the coupling between neighbors. We assume that each agent can communicate with all the neighbors to evaluate the shifted input trajectory as well as the global cost of the proposals. The weighting matrixes that define the cost function of the MPC controller are the following:

$$Q_i=\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix},\quad R_{ij}=10 \tag{63}$$

with $i=\{1,2,3,4\}$ and $j\in n_i$.

In order to implement the proposed DMPC control scheme we need to design the local feedbacks and the terminal cost functions according to LMI constraints presented in Theorem 2 to find matrices K_{ij} and P_i such that all the stability conditions are satisfied. In particular, matrices W, Y such that constraints (28) and (25) hold while maximizing the sum of the traces of the matrices W_i. Applying the variable change presented in Theorem 2, the following matrices K and P such that the stability assumptions hold are obtained

$$K^T=\begin{bmatrix}-0.27 & -0.01 & 0 & 0\\-0.59 & -0.02 & 0 & 0\\0 & -0.28 & 0 & -0.01\\-0.01 & -0.5 & 0 & -0.02\\0 & 0 & -0.24 & -0.01\\-0.01 & 0 & -0.68 & -0.02\\0 & -0.01 & 0 & -0.30\\0 & -0.02 & 0 & -0.48\end{bmatrix},$$
$$P=\begin{bmatrix}4.92 & 5.76 & 0 & 0 & 0 & 0 & 0 & 0\\5.76 & 11.30 & 0 & 0 & 0 & 0 & 0 & 0\\0 & 0 & 5.65 & 5.42 & 0 & 0 & 0 & 0\\0 & 0 & 5.42 & 8.82 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 4.45 & 5.81 & 0 & 0\\0 & 0 & 0 & 0 & 5.81 & 13.74 & 0 & 0\\0 & 0 & 0 & 0 & 0 & 0 & 5.61 & 5.80\\0 & 0 & 0 & 0 & 0 & 0 & 5.80 & 8.95\end{bmatrix} \tag{64}$$

The controller defined by matrix K stabilizes not only the centralized system but also the four subsystems individually considered. Note that in the optimization problem, additional constraints where imposed consisting in the absence of communication between some of the agents for the purpose of computing the local control law. This specification is reflected in the presence of zeros in the matrix. For example, agents 1 and 4 do not have to exchange any information in order to compute the shifted input trajectory. This class of additional constraints are particularly relevant when more involved communications protocols are taken into account.

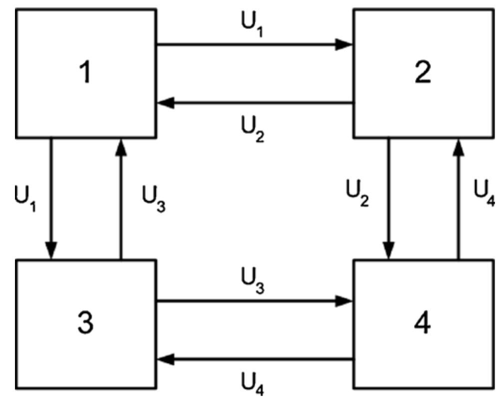

Fig. 4. Graph of the system (61).

The next step in the design procedure is to find the set of Λ that maximizes the size of the jointly invariant sets. In particular, we measure the size by the Chebyshev radius of the resulting centralized invariant set. The resulting optimization problem is convex problem (in particular, it can be posed as a LP problem) and has been solved using Matlab's `fmincon` function. The optimal matrix Λ is

$$\Lambda=\begin{bmatrix}0.4568 & 0.0931 & 0.0115 & 0\\0.0116 & 0.4576 & 0 & 0.0699\\0.0805 & 0 & 0.4908 & 0.0235\\0 & 0.1635 & 0.0354 & 0.4128\end{bmatrix} \tag{65}$$

where the element of the i-th row and the j-th column corresponds to the constant λ_{ij}. Note that the constants λ_{ji} that correspond to matrices $K_{ji}=0$ are set to zero.

The properties of the equivalent centralized system provide useful information to establish a comparison with the distributed approach. In particular, the size of the maximum invariant set for the centralized nominal case provides an upper bound of the size of the invariant set obtained from the jointly invariant sets. In this case, the radium of the largest Chebyshev ball is 0.74. The invariant set calculated for the distributed system has a radium of 0.66, a value very close to the centralized case. The reduction of the invariant region is 11%. In Fig. 5 the invariant set of each subsystem can be seen along with the corresponding projection of the centralized invariant set.

In general, the closed-loop stability properties are independent on how many proposals are evaluated or how this proposals are generated. This implies that the proposed controller scheme can be implemented using different proposal generation protocols. In this simulation, a communication protocol based on broadcast different from the one presented in Section 3 is used. At each sample time, each agent makes a single proposal optimizing its local cost function with respect to all the manipulated variables that affect him. All the proposals are compared (including U^s) and the one with the lower cost function is applied.

Fig. 7 shows the closed-loop state trajectories of all the subsystems with the corresponding jointly invariant sets. The simulations presented were done for a prediction horizon $N=12$, for the initial state

$$x_1(0)=\begin{bmatrix}-0.2311\\0.9072\end{bmatrix},\quad x_2(0)=\begin{bmatrix}-1.3558\\0.9929\end{bmatrix},$$
$$x_3(0)=\begin{bmatrix}-0.6533\\-0.2228\end{bmatrix},\quad x_4(0)=\begin{bmatrix}-1.0419\\1.1576\end{bmatrix} \tag{66}$$

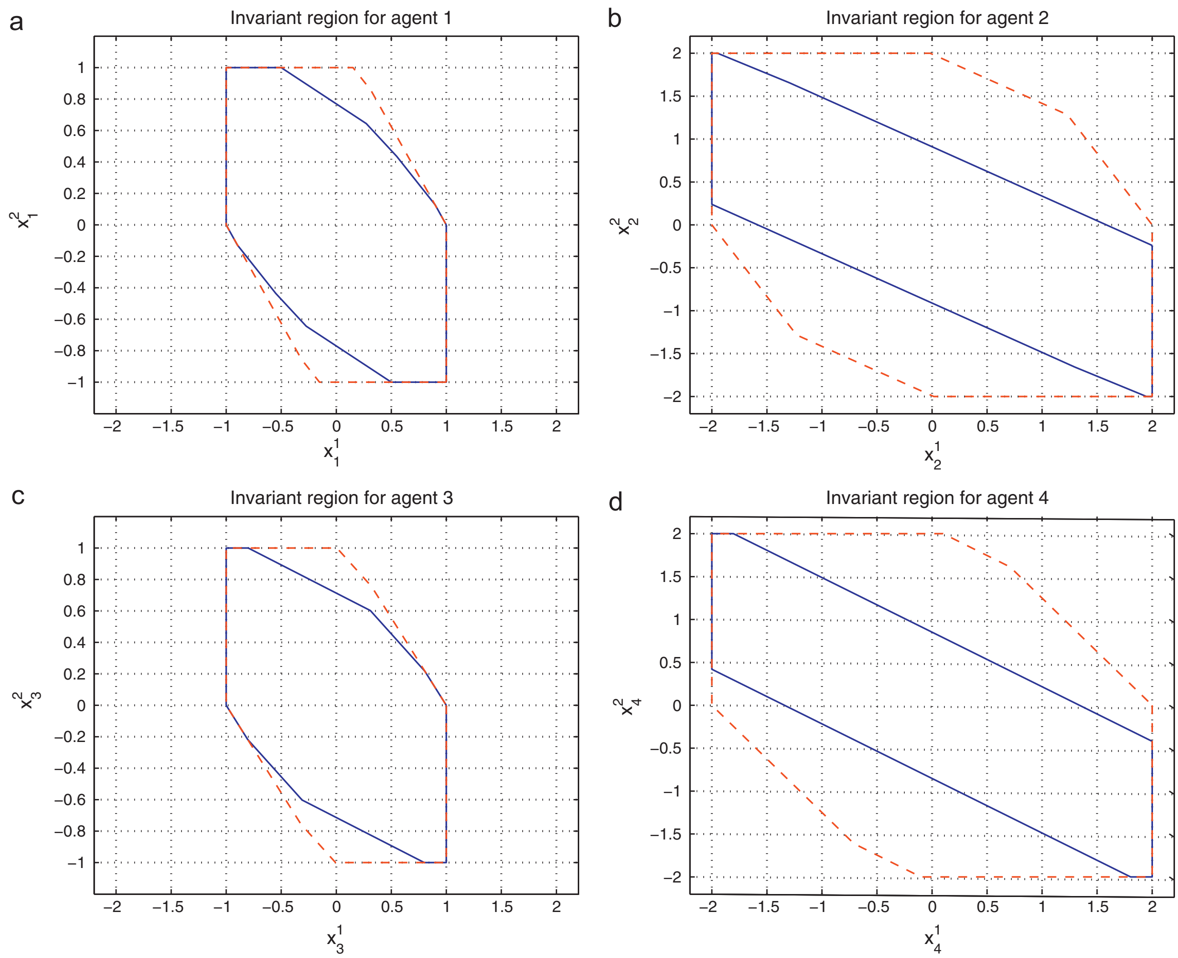

Fig. 5. Jointly invariant set of each subsystem (solid lines) along with the corresponding projection of the centralized invariant set (dashed lines).

and an initial control vector $U^s(0)$ calculated as a feasible control vector for the centralized system for such initial state.

Fig. 6 shows the proposal chosen at each time step. Numbers 1–4 indicate the agent that made the chosen proposal while 0 indicates that the shifted trajectory was chosen.

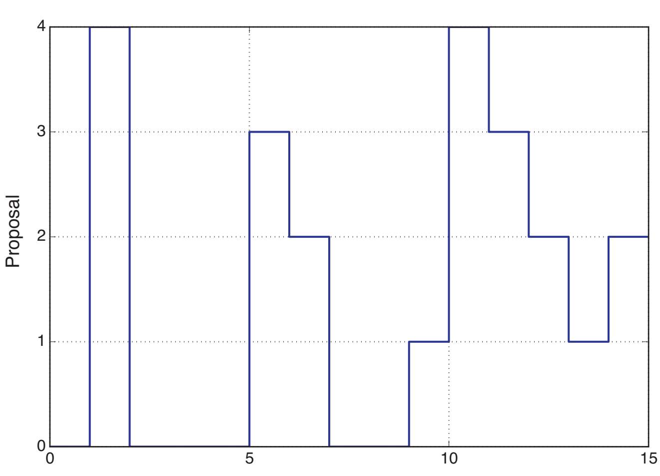

Fig. 6. Proposal chosen at each time step.

7. Application to a supply chain problem

In this section, we apply the proposed controller to a linear supply chain, which can be defined as the set of structures and processes used by an organization to provide a service or a good to a consumer. It is clear that the nodes of a supply chain may not have incentives to share other information about their models than their control actions. Supply chain flows usually present three interesting phenomena from the control point of view: oscillation, amplification and phase lag [39]. Due to material or informational delays production and inventories overshoot and undershoot the optimal levels. The magnitude of the fluctuations increase as they propagate from the customer to the factory, in what is commonly known as the bullwhip effect. For these reasons supply chains dynamics have been deeply analyzed and have been used as an application example in several distributed control papers [40,28].

In this paper, we consider a cascade of M_x firms. In particular, the discrete time equations that define the dynamics of firm i are given by:

$$s^i(t+1) = s^i(t) + u^{i-1}(t - d_{i-1,i}) - u^i(t) \qquad (67)$$

The super-scripts $i-1$ and $i+1$ represent, respectively, the dynamics of the upstream and downstream nodes. Variable $s^i(t)$ is the stock level; that is, the number of items available for shipment downstream. The manipulated variable at each stage is $u^i(t)$ which stands for the number of items sent to the downstream node. This is a difference with respect to models in which there is one variable

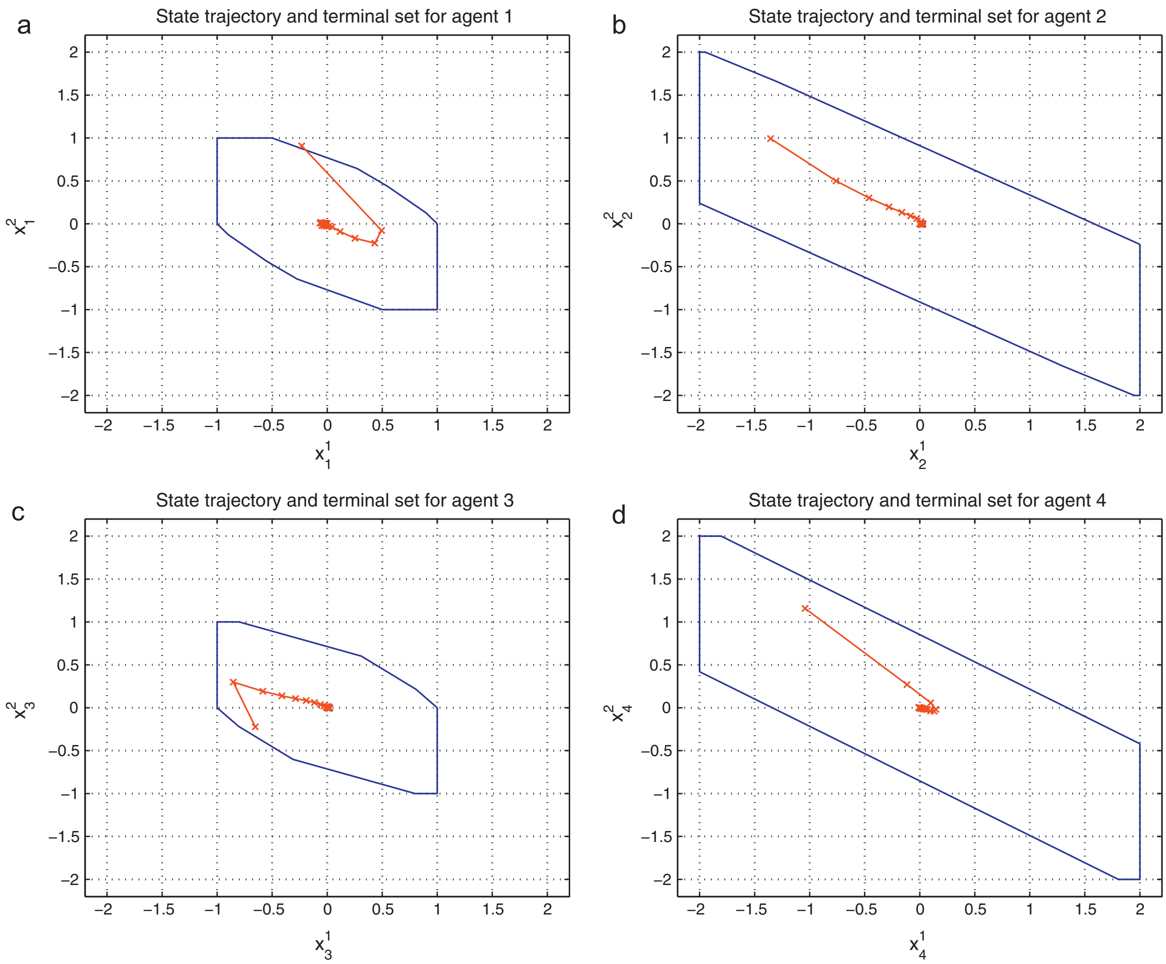

Fig. 7. (a) Agent 1 state evolution. (b) Agent 2 state evolution. (c) Agent 3 state evolution. (d) Agent 4 state evolution.

that stands for the order rate and another, which is usually modeled as a disturbance, that stands for the shipment itself. The information flows are assumed to have no time delays and the material flows have a delay modeled by $d_{i,j}$ which corresponds to the time taken by the shipments from node i to node j (Fig. 8).

The only information shared by the agents is their inputs. In particular, the model of a node needs to keep track of the shipments made by its upstream node. This implies that a model of the form (1) can be obtained assigning a different subsystem to each firm i with the following state vector x_i

$$x_i(t) = \begin{bmatrix} s^i(t) \\ u^{i-1}(t-1) \\ u^{i-1}(t-2) \\ \vdots \\ u^{i-1}(t-d_{i-1,i}) \end{bmatrix}$$

Note that this model takes into account the different delays by augmenting the state vectors. The inputs are defined by the different shipments variables u^j.

In this model the first firm, with state $x_1(t)$ is the supplier which demands items directly to the factory by $u_0(t)$ which is modeled as a pure delay of value $d_{0,1}$. The last firm is the retailer which must satisfy the external demand $u_{M_x}(t)$ which is an external signal not controlled by the system. The control objective is to regulate the stock levels to a desired value $r^i(t)$. In addition, the last node of supply chain, the retailer, has to satisfy the external demand.

To this end, we consider the following local cost function for each firm

$$J_i = \sum_{k=1}^{N} 2^i \left(r_k^i - s_k^i - \sum_{l=k-1}^{d_{i-1,i}} u_{k-l}^{i-1} \right)^2$$

where N is the prediction horizon, the subindex k denotes the k-steps predicted value of a signal. No terminal cost function is considered. The cost penalizes the deviation of the sum of current stock and the items traveling from the upstream node from the desired reference. Note that if the controller ignores those units that have to arrive in the future, it would ask for more units than needed. The weights of the local cost grow with 2^i, that is, the closer a node

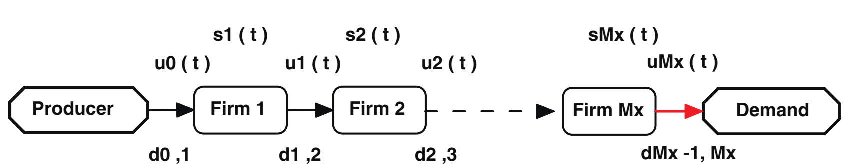

Fig. 8. Linear supply chain.

is to the retailer the more important is. This way of weighting the error is natural since the most important goal of a supply chain is to satisfy the external demand.

The class of linear supply chains considered in this example is defined by the number of firms M_x and the delay parameters $d_{i,j}$. In the following tables we show the results of a set of simulations with three different supply chains of 5, 10 and 20 firms. We denote these scenarios as SUPPLY5, SUPPLY10 and SUPPLY20 respectively. The delay parameters of each supply chain have been randomly chosen with values between 2 and 5. The initial stock $s^i(0)$ was chosen randomly between 100 and 300. In all these simulations, we assume that the external demand $u_{M_x}(t)$ is null and that the objective of the controller is to regulate the stocks to their references. The references $r^i(t)$ were supposed to be constant and were chosen randomly between 180 and 280. The simulation times T_f were set respectively to 50, 100 and 200 sample times.

In order to study the effect of the number of proposals considered at each sampling time in the performance of the proposed DMPC scheme, we have applied several controllers which consider a different number of proposals N_{prop}. Given that the proposals are made randomly, each simulation was repeated 10 times. In addition, a centralized MPC controller has also been applied to the three scenarios as a reference of the performance that can be obtained with a centralized approach.

Tables 1–3 show the cumulated cost mean $\bar{J}_{cum}$ and the corresponding standard deviation σ_J of each controller. The cumulated cost of each simulation was computed as:

$$J_{cum} = \sum_{t=0}^{T_f} \sum_{i=1}^{M_x} 2^i (r^i(t) - s^i(t))^2$$

In addition the tables show the mean number of sample times $\bar{t}_{ss}$ that an agent needs to have less than a 5% of error with respect to its reference, as well as the average number of sample times t_{ss} that the slowest agent needs to have less than a 5% of error with respect to its reference. These two entries provide additional information on the performance of each controller.

Table 1
Simulation results for SUPPLY5.

Controller	$\bar{J}_{cum}$	σ_J	$\bar{t}_{ss}$	t_{ss}	N_{prop}
DMPC	2.36e+6	1.80e+6	25.57	30.70	1
DMPC	9.39e+5	2.99e+6	15.20	18.30	3
DMPC	6.64e+5	3.05e+5	11.25	13.80	5
DMPC	5.53e+5	1.40e+5	9.85	12.20	7
DMPC	5.39e+5	1.69e+5	9.05	11.30	10
DMPC	4.34e+5	8.30e+4	8.25	10.50	15
DMPC	3.88e+5	1.25e+4	7.70	9.50	20
DMPC	3.86e+5	1.24e+4	7.65	9.30	30
MPC	3.71e+5	–	7.50	9.00	–

Table 2
Simulation results for SUPPLY10.

Controller	$\bar{J}_{cum}$	σ_J	$\bar{t}_{ss}$	t_{ss}	N_{prop}
DMPC	8.50e+7	1.95e+7	58.32	93.50	1
DMPC	4.57e+7	1.20e+7	29.48	46.30	3
DMPC	2.61e+7	3.28e+6	21.78	34.30	5
DMPC	2.62e+7	4.23e+6	20.34	29.50	7
DMPC	2.06e+7	2.98e+6	16.18	24.20	10
DMPC	1.71e+7	1.98e+6	13.81	21.30	15
DMPC	1.70e+7	2.00e+6	13.68	21.50	20
DMPC	1.63e+7	1.25e+6	12.82	20.50	30
DMPC	1.53e+7	9.56e+5	12.91	20.50	50
DMPC	1.52e+7	7.07e+5	12.36	20.10	70
DMPC	1.52e+7	6.51e+5	12.07	20.10	100
MPC	1.45e+7	–	13.00	20.00	–

Table 3
Simulation results for SUPPLY20

Controller	$\bar{J}_{cum}$	σ_J	$\bar{t}_{ss}$	t_{ss}	N_{prop}
DMPC	5.84e+11	1.24e+11	162.49	199.80	1
DMPC	2.66e+11	6.02e+10	132.96	187.00	3
DMPC	1.46e+11	2.83e+10	93.34	137.50	5
DMPC	1.30e+11	1.48e+10	76.09	112.90	7
DMPC	9.85e+10	1.73e+10	60.21	88.90	10
DMPC	8.23e+10	1.11e+10	49.48	73.40	15
DMPC	6.19e+10	7.09e+9	42.24	61.70	20
DMPC	5.55e+10	4.52e+9	39.02	56.40	30
DMPC	5.24e+10	3.01e+9	32.38	46.70	50
DMPC	5.07e+10	1.38e+9	31.15	44.30	70
DMPC	5.01e+10	9.34e+8	30.36	42.90	100
DMPC	4.98e+10	8.90e+8	29.91	42.10	125
DMPC	4.97e+10	5.79e+8	29.65	41.80	150
DMPC	4.98e+10	4.03e+8	29.23	41.90	175
DMPC	4.95e+10	6.09e+8	28.31	41.10	200
MPC	3.84e+10	–	19.53	26.00	–

In general, the simulations show that increasing the number of proposals N_{prop} improves the performance of the proposed DMPC scheme. It can be seen that $\bar{J}_{cum}$ and σ_J are decreasing functions of the parameter N_{prop}. However, communications can be a scarce resource for some systems and it is important to find a trade-off between the number of communications and the performance. In our example it can be seen that a good trade-off happens when N_{prop} is around $5M_x$ communications, where M_x is the number of agents. This implies that each agent makes an average of 5 proposals to its neighbors.

8. Conclusions

In this work we presented a novel distributed MPC algorithm based on negotiation for a class distributed linear systems coupled through the inputs. We assume that each agent has access only to the model and the state of one of the subsystems and that the agents must communicate in order to reach a cooperative solution. The proposed algorithm provides a negotiation protocol on behalf of a global cost function that can be implemented in parallel and provides a feasible solution to the centralized problem. In addition, we provide sufficient conditions that guarantee asymptotic stability of the closed-loop system as well as an optimization based procedure to design the controller so that these conditions are satisfied.

Acknowledgements

Financial support from the European Commission, INFSO-ICT-223854 and MEC-Spain, DPI2008-05818, is gratefully acknowledged.

References

[1] R. Negenborn, B. De Schutter, H. Hellendoorn, Multi-agent model predictive control of transportation networks, in: Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control (ICNSC 2006), Ft. Lauderdale, Florida, 2006, pp. 296–301.
[2] T.C. Yang, Networked control systems: a brief survey, in: IEE Procedings – Control Theory and Applications, Vol. 152, 2006, pp. 403–412.
[3] P. Neumann, Communication in industrial automation – what is going on? Control Engineering Practice 15, 1332–1347.
[4] T. Basar, G.J. Olsder, Dynamic Noncooperative Game Theory, Society for Industrial and Applied Mathematics, Philadelphia, 1999.
[5] R.B. Myerson, Game Theory. Analysis of Conflict, Hardvard University Press, 1997.
[6] H.R. Varian, Intermediate Microeconomics: A Modern Approach, W.W. Norton & Co., 2005.
[7] E. Camacho, C. Bordons, Model Predictive Control in the Process Industry, second ed., Springer-Verlag, London, England, 2004.
[8] J.B. Rawlings, B.T. Stewart, Coordinating multiple optimization-based controllers: new opportunities and challenges, Journal of Process Control 18 (9) (2008) 839–845.

[9] R. Scattolini, Architectures for distributed and hierarchical model predictive control – a review, Journal of Process Control 19 (2009) 723–731.
[10] A. Alessio, A. Bemporad, Decentralized model predictive control of constrained linear systems, in: Proceedings of the 2007 European Control Conference, 2007, pp. 2813–2818.
[11] A. Alessio, A. Bemporad, Stability conditions for decentralized model predictive control under packet drop communication, in: Proceedings of the 2008 American Control Conference, 2008, pp. 3577–3582.
[12] D. Jia, B. Krogh, Distributed model predictive control, in: Proceedings of the 2001 American Control Conference, 2001, pp. 4507–4512.
[13] D. Jia, B. Krogh, Min-max feedback model predictive control for distributed control with communication, in: Proceedings of the American Control Conference, Anchorage, 2002, pp. 4507–4512.
[14] A.N. Venkat, J.B. Rawlings, S.J. Wright, Stability and optimality of distributed model predictive control, in: Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference ECC 2005, Seville, Spain, 2005, pp. 6680–6685.
[15] M. Mercangoz, F.J. Doyle, Distributed model predictive control of an experimental four-tank system, Journal of Process Control 17 (2007) 297–308.
[16] A. Rantzer, Dynamic dual decomposition for distributed control, in: American Control Conference, 2009.
[17] R.R. Negenborn, P.J. Van Overloop, T. Keviczky, B. De Schutter, Distributed model predictive control of irrigation canals, Networks and Heterogeneus Media 4 (2009) 359–380.
[18] E. Camponogara, B. Barcelos de Oliveira, Distributed optimization for model predictive control of linear-dynamic networks, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 39 (2009) 1331–1338.
[19] H. Scheu, J. Busch, W. Marquardt, Nonlinear distributed dynamic optimization based on first order sensitivities, in: Proceedings of the American Control Conference, Baltimore, Maryland, USA, 2010.
[20] J. Liu, D. Muñoz de la Peña, B. Ohran, P.D. Christofides, J.F. Davis, A two-tier architecture for networked process control, Chemical Engineering Science 63 (2008) 5349–5409.
[21] J. Liu, D. Muñoz de la Peña, B.J. Ohran, P.D. Christofides, J.F. Davis, A two-tier control architecture for nonlinear process systems with continuous/asynchronous feedback, International Journal of Control 83 (2) (2010) 257–272.
[22] J. Liu, D.M. noz de la Peña, P.D. Christofides, Distributed model predictive control of nonlinear process systems, AIChE Journal 55 (2009) 1171–1184.
[23] J. Liu, D. Muñoz de la Peña, P.D. Christofides, Distributed model predictive control of nonlinear systems subject to asynchronous and delayed measurements, Automatica 46 (1) (2009) 52–61.
[24] D.P. Bertsekas, J.N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, Athena Scientific, 1997.
[25] A.N. Venkat, Distributed model predictive control: Theory and applications, Ph.D. thesis, University of Wisconsin-Madison, 2006.
[26] B.T. Stewart, A.N. Venkat, J.B. Rawlings, S.J. Wright, G. Pannocchia, Cooperative distributed model predictive control, Systems and Control Letters.
[27] J.B. Rawlings, D.Q. Mayne, Model Predictive Control: Theory and Design, Nob Hill Publishing, 2009.
[28] J.M. Maestre, D. Muñoz de la Peña, E.F. Camacho, Distributed model predictive control based on a cooperative game, Optimal Control Applications and Methods, in press, doi:10.1002/oca.940.
[29] D.Q. Mayne, J.B. Rawlings, C.V. Rao, P.O.M. Scokaert, Constrained model predictive control: stability and optimality, Automatica 36 (2000) 789–814.
[30] M. Lazar, W.P.M.H. Heemels, S. Weiland, A. Bemporad, Stabilizing model predictive control of hybrid systems, IEEE Transactions on Automatic Control 51 (2006) 1813–1818.
[31] P.O.M. Scokaert, D.Q. Mayne, J.B. Rawlings, Suboptimal model predictive control (feasibility implies stability), IEEE Transactions on Automatic Control 44 (1999) 648–654.
[32] M. Kothare, V. Balakrishnan, M. Morari, Robust constrained model predictive control using linear matrix inequalities, Automatica 32 (1996) 1361–1379.
[33] T. Alamo, D. Muñoz de la Peña, D. Limon, E. Camacho, Constrained min-max predictive control: Modifications of the objective function leading to polynomial complexity, IEEE Transactions on Automatic Control 50 (5) (2005) 710–714.
[34] M. Lazar, W.P.M.H. Heemels, D. Muñoz de la Peña, T. Alamo, Further results on "robust mpc using linear matrix inequalities", in: Proceedings of Int. Workshop on Assessment and Future Directions of NMPC, Pavia, Italy, 2008.
[35] S.V. Rakovic, E. De Santis, P. Caravani, Invariant equilibria of polytopic games via optimized robust control invariance, in: Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference, Seville, Spain, 2005, pp. 7686–7691.
[36] I. Kolmanovsky, E.G. Gilbert, Theory and computation of disturbance invariant sets for discrete-time linear invariant systems, Mathematical Problems in Engineering: Theory, Methods and Applications 4 (1998) 317–367.
[37] S. Rakovic, E. Kerrigan, K. Kouramas, D. Mayne, Robust mpc for nonlinear discrete-time systems, IEEE Transactions on Automatic Control 50 (2005) 406–410.
[38] E.C. Kerrigan, Robust constraint satisfaction: Invariant sets and predictive control, Ph.D. thesis, Department of Engineering, University of Cambridge, UK, November 2000.
[39] J.D. Sterman, Business Dynamics – Systems Thinking and Modelling in a Complex World, McGraw-Hill, New York, 2000.
[40] W.B. Dunbar, S. Desa, Distributed mpc for dynamic supply chain management, in: International Workshop on Assessment and Future Directions of NMPC, Freudenstadt-Lauterbad, Germany, 2005, pp. 26–30.